

Design of a New Telescope Control System for Use in Astronomical Transient Events

Selcuk Helhel^{*1}, Murat Dindar², Ekrem Kandemir², Saniye Dindar², Tuncay Ozişik²

¹ Department of E.E.E., Engineering Faculty, Akdeniz University, 07058, Antalya, Turkey

²TUBITAK National Observatory, Akdeniz University Campus, 07058, Antalya, Turkey

*Corresponding author, email: selcukhelhel@akdeniz.edu.tr, murat.dindar@tubitak.gov.tr, ekrem.kandemir@tubitak.gov.tr, saniye.dindar@tubitak.gov.tr, tuncay.ozisik@tubitak.gov.tr

Abstract

Robotic autonomous telescopes provide high level control by selecting astronomical targets for observation, and they usually run under the control of a scheduler. TalonVIEW is a newly designed robotic autonomous telescope control system (TCS) for 16 inches telescope mount for use in astronomical transient events. The telescope control algorithm was implemented in PXI chassis written in GProgramming (LabVIEW) on real-time operating system (PharLab) from scratch. A new TCP/IP library was also implemented in Talon software to communicate with Pharlab in PXI chassis. Initial setup without any permanent pier and polar alignment showed that the pointing error of the telescope has been obtained as 2.22 arcminutes (132 arc seconds) in RA axis (horizontal axis of the image) and 25 arc seconds in Dec axis (vertical axis of the image), and tracking error has been observed as 4.8 arc seconds per second.

Keywords: *Autonomous Robotic Telescopes, Astronomical Transient Events, G-Programming Telescope Control Software.*

Copyright © 2016 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Computer-based robotic telescope systems have the advantages of low operating costs with high operating efficiency and high scientific productivity [1, 2]. Some demonstrations of scientific usage of the robotic telescope were given in [1-5]. Generally the term robotic in telescope systems stand for guiding a telescope to a given position and take images or perform more complicated tasks. The term autonomous observatory in astronomy means a robotic telescope and dome are computer controlled in such manner, that all indispensable actions of observation are done automatically, including processing of weather conditions, dome driving, choosing objects to observe, exposing by cameras or other optical sensors, taking calibration images, and so forth. Neither human interaction nor activity is necessary for observation [5]. Removing human control from the observing process allows faster observation response time, so that it makes robotic telescopes respond quickly to alert broadcasts from satellites and begin observing within seconds. This property lets the astronomers to observe transient events (Gamma-Ray Bursts etc.) in the sky.

Telescope control algorithm can be executed on various platforms where LabVIEW runtime is available, among them; Windows and Linux, real-time extension for Windows (RTX), and real-time operating systems (PharLab ETS, VxWorks etc.). Some telescope control software applications on Lab-VIEW platform have also been given in [6, 12].

Talon, observatory control software (OCS), have already been improved and integrated to new systems at TUBITAK National Observatory (TUG) by using advantages of its open-source software structure [13, 14]. This paper presents a newly designed TCS and adapted to Talon in order to control the telescope mount, named TUY 40, to be an autonomous robotic telescope.

2. TUY 40 Autonomous Robotic Telescope

TUY 40 telescope mount is controlled by a new TCS written in G-Programming called as TalonVIEW, which is implemented in PXI real-time system. The control architecture of TUY 40 is simply shown in Figure 1.

2.1 Hardware Architecture

Motion control algorithm for the main axes of RA and Dec was implemented within motor drivers and motion controller cards. Two servo motors were used to move the axes. Precise motion control is a prerequisite for a telescope. Therefore, closed loop PID control was implemented for accurate position control within high resolution motor drivers with encoders (Figure 2). Motion controller cards reliably execute programs with very precise timing. They are playing the key role to deliver the target position and velocity values sent by the software to the drivers. Brand new designs with different actuators and with different equipment could be easily adopted by the telescope control system without any modification in the code because of the excellent compatibility of the motion control cards. Therefore, in the long run, hardware architecture sets the stage for further developments.

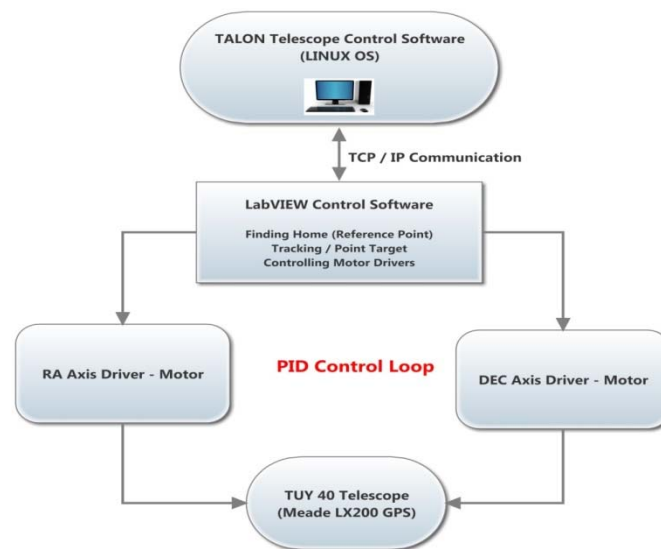


Figure 1. The control architecture of TUY 40

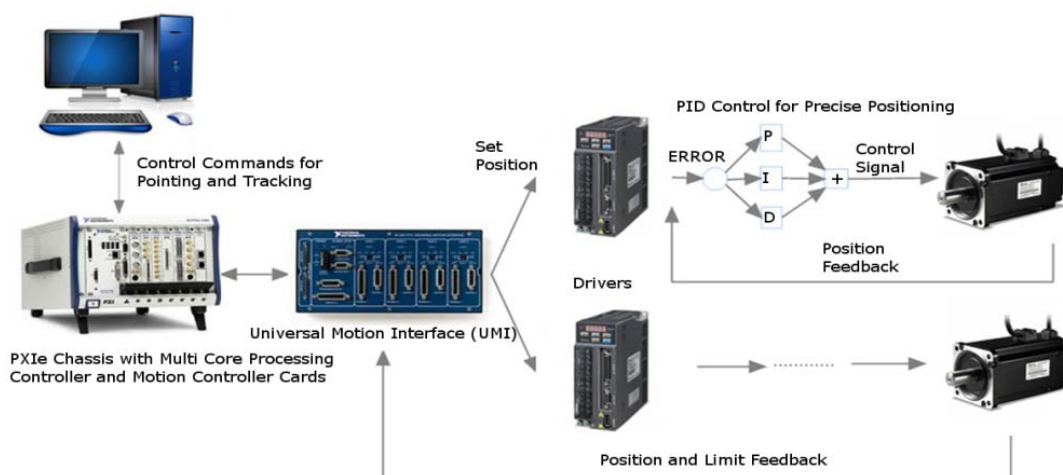


Figure 2. Hardware architecture of TUY40 telescope

The commands coming from the computer is evaluated by Real Time Embedded Controller. Then it sends appropriate values to the motion controller cards. Required pulse and direction voltages are received by servo drivers as target position value and they start the motion of the axes. Then it reads encoder feedback signals and runs the PID control loop to carry out precise positioning within specified time interval. When it accomplishes accurate

pointing then it starts tracking in order to enable the telescope to observe the target object with a defined speed value. Limit switches are used to achieve safe movement of telescope. Motion controller cards have precaution action. As soon as they receive a limit switch signal, they stop the telescope movement immediately.

2.2 Software Architecture

The new TCS control software called as TalonVIEW. The new architecture is simply based on the Client/Server model. The client side of new TCS software is based on Talon, which is an open-source observatory control software and written in GNU/C [15], with a newly written TCP/IP library and application functions of this library. It is used to make astronomical calculations, control camera and observatory modules such as GPS, meteo station etc. Talon has also been using in T60 autonomous telescope in TUG in different software and hardware architecture [13, 14]. Talon has two main daemons of telrun and telescoped.

The telrun daemon is responsible for operating the schedule on robotic mode by sending appropriate command to telescope daemon (telescoped). The telescoped is responsible for the control of the dome, focus, filter wheel and the telescope axes by sending the low-level commands to the motion controllers. The motion controller cards can have been programmed by using built-in functions.

This new architecture makes the Talon dependent on the specific motion controllers. The challenge is to integrate Talon to more robust platform than already used. The server side of new TCS software is written in G-programming which is running on the PXI chassis with real-time embedded controller and motion controller cards.

Both Talon and TalonVIEW GUI are shown in Figure 3 and Figure 4. All subVIs are written from scratch in order to control the subsystems such as telescope axes, filter, focus and dome. The subVIs called by main VI which interprets and parses the commands from client side (Talon) and processes all the commands in real-time and simultaneously by using the multithread feature of G-programming. The real-time queue functions are used to provide the synchronous of the processes in Real-Time Operating System.

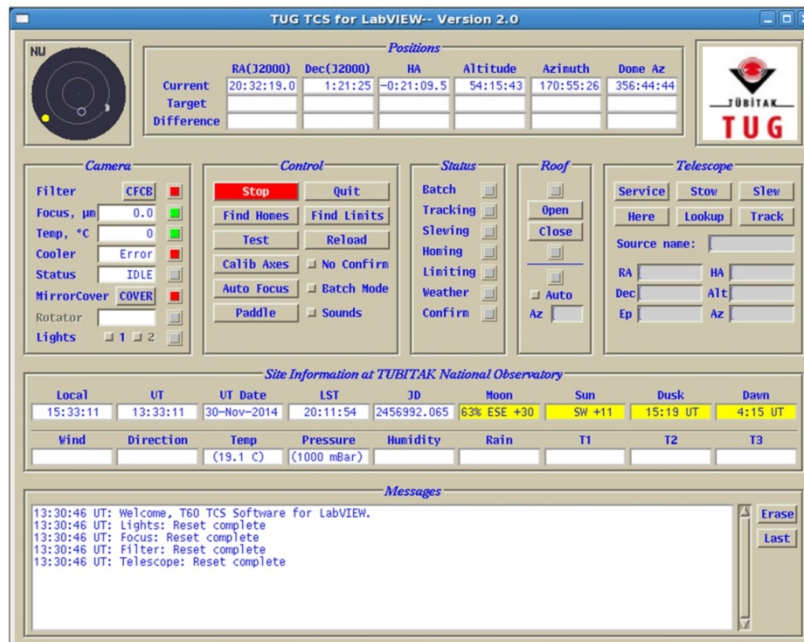


Figure 3. TALON GUI

All subsystems on server side are identical to subsystems on client side. They have a special command tags to send the commands to the related modules on server and read relevant data with specified tags received from the client, thus it is guaranteed that the received

data from server is taken by subsystems from client correctly. The architecture of whole TUY 40 TCS (Talon and TalonVIEW) are shown in Figure 5.

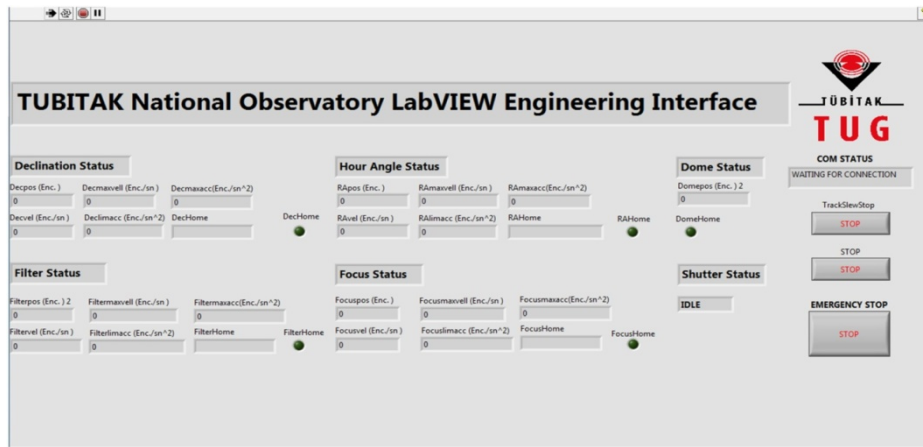


Figure 4. TALON VIEW GUI

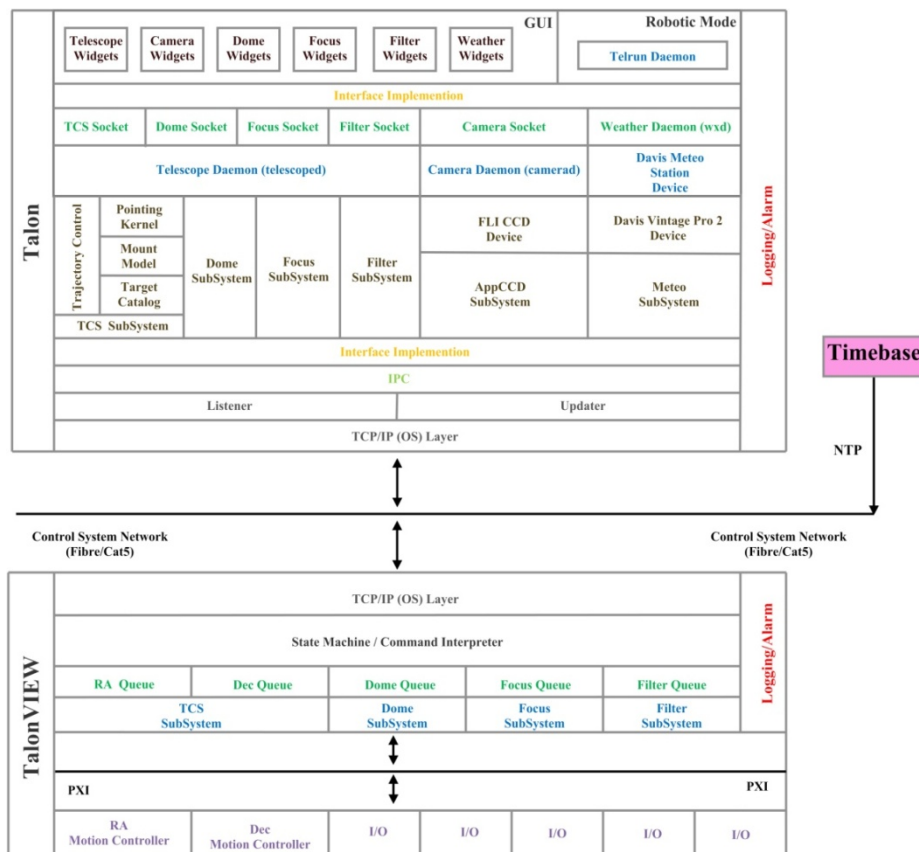


Figure 5. The software architecture of TUY 40 TCS

3. Experimental Setup

The experimental setup of TUY 40 autonomous telescope system is shown in Figure 6. Commercially present optical tube was used for the sky tests of the telescope. The sky tests were realized in order to take pointing and tracking errors of TUY 40. For the experimental

results of the TUY 40, using very rough polar alignment without permanent pier, Canon EOS 20 Da Digital SLR 100 Camera was used as image detector system on the telescope focal plane. The camera has 3504 x 2336 pixel area. Corresponding to this pixel area, field of view of the camera is 19.1 x 12.8 arcmin² and pixel scale of the image is 0.32 arc seconds per pixel.

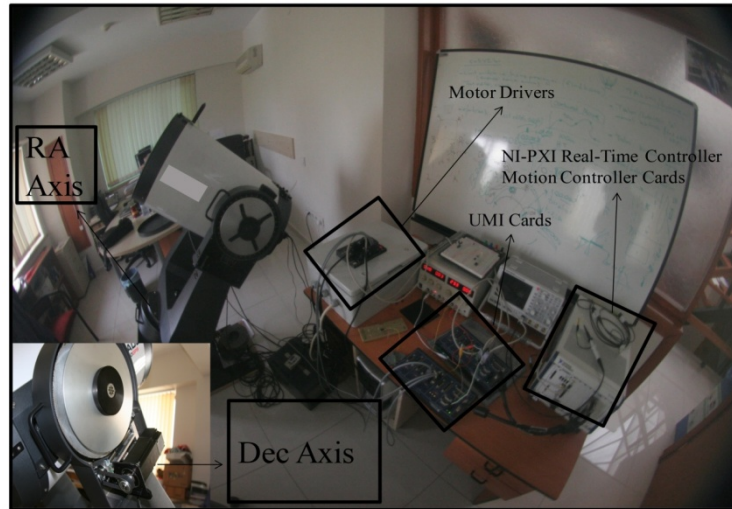


Figure 6. Experimental setup of TUY40 telescope

Target object (star) is shown in a white border square area of the image. The red-cross shows location where the telescope pointed and the green cross shows real location of the star as shown in Figure 7. So, the difference between these points gives telescope pointing error. The pointing error of the telescope is 2.22 arcminutes (132 arc seconds) in RA axis (horizontal axis of the image) and 25 arc seconds in Dec axis.

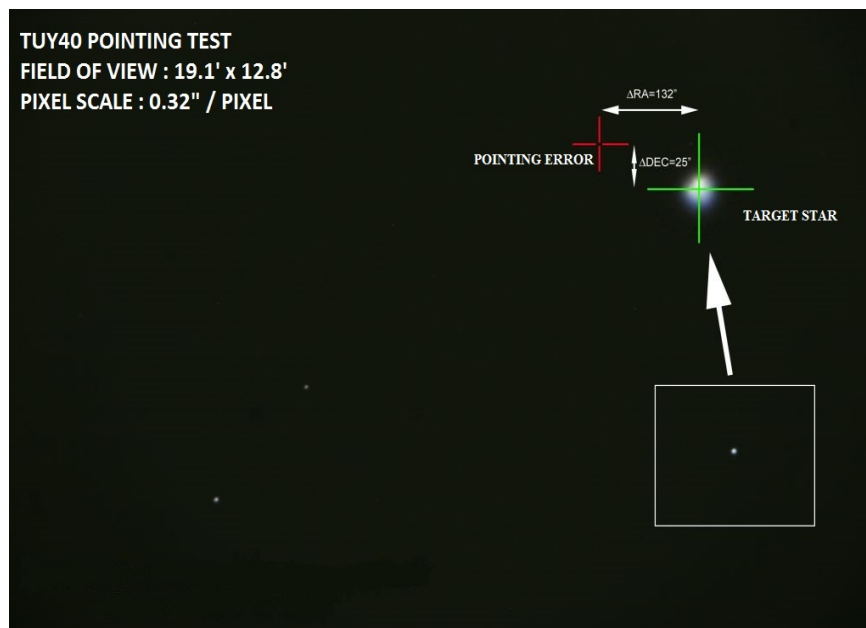


Figure 7. Experimental result of TUY40 telescope for pointing

Tracking tests are realized on the sky, as well. For the tracking tests, three types of images was taken, 20s, 60s and 100 s. As a result of analyzing of those images, tracking error was found as 4.8 arc seconds per second approximately. These test results were obtained in experimental setup conditions where the mount of the telescope was not installed on robust platform such as permanent pier or strong base. After the installation of the telescope and make pointing model on sky will significantly improve the tracking and pointing performance of telescope in near future.

4. Conclusion

In this study; TalonVIEW, which is a new robotic autonomous TCS, has been designed on PharLab, Real-Time Operating System, and implemented within a real time platform by integrating to Talon. Satisfying results have been obtained for pointing and tracking in experimental setup. For upcoming studies, telescope will be installed on a robust platform where will be located at Bakirlitepe, Antalya (Lat: 36° 54' 00" N., Long: 30° 39' 13" E.) so that pointing model of the telescope will be improved. The new software (altrtd), which runs under the Talon, have been used for astronomical transient events in T60 telescope at TUG [14]. After the commissioning of TUY 40, the telescope will be dedicated to catch for astronomical transient events such as supernovae, gamma-ray bursts etc. by integrating the altrtd to the TCS.

Acknowledgements

This project was granted by State Planning Organization (DPT Grant Number: 2010K120170), and we would like to thank Mr. Cevdet Bayar for his kind cooperation.

References

- [1] Boyd LJ, Genet RM, Hall DS. APT's - Automatic photoelectric telescopes. *Sky and Telescope*, 1985;70: 16-19.
- [2] Drummond M, Bresina J, Edgington W, Swanson K, Henry G, Drascher E. *Robotic Telescopes: Current Capabilities*. Present Developments, and Future Prospects for Automated Astronomy, Pub. Astron. Soc. Pac. San Francisco. 1995: 70-112.
- [3] Smith AB: The development and implementation of a remote robotic telescope system at Appalachian State University's Dark Sky Observatory. M.D. dissertation. North Carolina: Physics and Astronomy, Appalachian State University; 2009.
- [4] Ferrero A, Hanlon L, Felletti R, French J, Melady G, McBreen S, Kub_aneek P, Jelinek M, McBreen B, Meintjes P, Calitz J, Ho_man M. The photometry pipeline of the Watcher Robotic Telescope. *Advances in Astronomy*. 2010.
- [5] Nekola M, Hudec R, Jel_nek M, Kub_aneek P, trobl J, Pol_a_sek C. BART: The Czech Autonomous Observatory. *Advances in Astronomy*. 2010.
- [6] Zagar K, Zagar A, Bauvir B, Chiozzi G, Duhoux P. *Integration of ALMA common software and national instruments LabVIEW*. Proceedings of PCaPAC08. Ljubljana. 2008.
- [7] Michael A, Schumacher G. *SOAR telescope control system: a rapid prototype and development in LabVIEW*. Proc. of SPIE. Munich. 2009; 4009.
- [8] Michael A, Marco B, Steve H. *ArcVIEW: a LabVIEW-based astronomical instrument control system*. Proc. of SPIE. Hawaii. 2002; 4848.
- [9] Schumacher G, Michael A. *SOAR TCS: from prototype to implementation*. Proc. Of SPIE. Hawaii. 2002; 4848.
- [10] Taylor P. *Telescope control using LabVIEW for Linux*. NI Big Physics Symposium. Rome. 2010.
- [11] Paul JL, Daniel G, Ryan G, Philip T. *Discovery Channel Telescope software development overview*. Proc. of SPIE. California. 2010; 7740.
- [12] Gilles F, Shan M, Philippe L, Abdel A, Gilles B, Isabelle J. *Innovative telescope control system architecture for SST-GATE telescopes at the CTA Observatory*. Proc. of SPIE. Montral. 2014; 9145.
- [13] Parmaksizoglu M, Dindar M, Helhel S, Kirbiyik H. *Software and Electronic Developments for TUG-T60 Robotic Telescope*. Conf. Series of Rev. Mex. de Astronomia y Astrofisica. 2014; 45: 24-25.
- [14] Dindar M, Helhel S, Esenonglu H, Parmaksizoglu M. A New Software on TUGT60 Autonomous Telescope for Astronomical Transient Events. *Experimental Astronomy*. 2015; 39: 21-28.
- [15] Reference Manual for Talon (OCAAS) Version 2.0. 2000.